VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

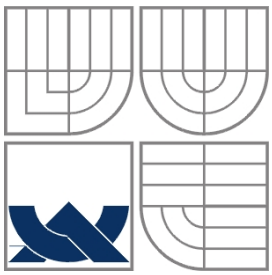
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VYHLEDÁVÁNÍ SPECIFICKÝCH SEKUNDÁRNÍCH STRUKTUR V DNA SEKVENCÍCH

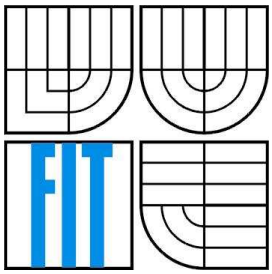
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JIŘÍ NĚMEC
AUTHOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VYHLEDÁVÁNÍ SPECIFICKÝCH SEKUNDÁRNÍCH STRUKTUR V DNA SEKVENCÍCH

FINDING SPECIFIC SECONDARY STRUCTURES IN DNA SEQUENCES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ NĚMEC

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MARTÍNEK

BRNO 2010

Abstrakt

Tato bakalářská práce se zabývá analýzou a návrhem algoritmu pro vyhledávání specifických struktur v DNA sekvencích, konkrétně vyhledávání kvadruplexů. Hlavním cílem práce bylo navrhnout a implementovat efektivní aplikaci s lineární časovou složitostí. Práce zahrnuje popis problematiky vyhledávání a použitých algoritmů, včetně jejich složitosti.

Abstract

This Bachelor thesis is focused on analyse and design an algorithm for searching specific structure in DNA sequences, especially for detection of quadruplexes. Work objective is to design and implement an effective application with linear time complexity. Work includes description of searching and used algorithms including their complexity.

Klíčová slova

DNA, sekundární struktura, kvadruplex

Keywords

DNA, secondary structure, quadruplex

Citace

Němec Jiří: Vyhledávání specifických struktur v DNA sekvencích, bakalářská práce, Brno, FIT VUT v Brně, 2010

Vyhledávání specifických struktur v DNA sekvencích

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Tomáše Martínka.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jiří Němec
18.5.2010

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Tomáši Martínkovi za vedení práce, rady a trpělivost.

Obsah

1 Úvod

Nukleové kyseliny jsou základní makromolekulární látkou živých soustav a jsou nositelkami genetické informace. Díky těmto nukleovým kyselinám dochází k přenosu dědičných znaků na potomstvo. Genetická informace je uložena v DNA, která má tvar dvojité šroubovice. Kromě této klasické formy ale existují i sekundární struktury, které mají jiný tvar. Mezi příklady těchto struktur patří kvadruplexy, což jsou sekundární struktury, které se skládají ze čtyř řetězců. V současnosti roste zájem o tyto struktury a probíhá intenzivní výzkum, neboť kvadruplexy hrají důležitou úlohu v biologických procesech. Tato oblast není ještě příliš prozkoumaná a o jejich biologické roli se moc neví. Kvadruplexy jsou zajímavé pro svůj velký potenciál jak v medicíně, tak i v nanotechnologii. Pokud chceme kvadruplexy zkoumat laboratorně, je to velmi složité. Proto se zde nabízí možnost využít výpočetní techniky, která dokáže vyhledávat potenciální výskyty kvadruplexů.

Tato bakalářská práce je zaměřena především na návrh a implementaci algoritmu, který bude schopen tyto specifické sekundární struktury vyhledat.

První část této práce se věnuje teoretickému úvodu, kde je přiblíženo, co je to DNA, sekundární struktura a podrobněji jsou zde rozebrány kvadruplexy. V další kapitole je popsána problematika vyhledávání kvadruplexů a vysvětleno vyhledávací pravidlo. Čtvrtá kapitola se zabývá analýzou a návrhem programu. V páté kapitole jsou rozepsány jednotlivé algoritmy, jejich složitost a také použité datové struktury. V části týkající se testování jsou uvedeny zajímavé výsledky, které byly následně zjištěny. Závěr obsahuje zhodnocení dosažených výsledků a vlastní přínos. Také jsou zde nastíněny další možnosti vývoje do budoucna.

2 Teoretický úvod

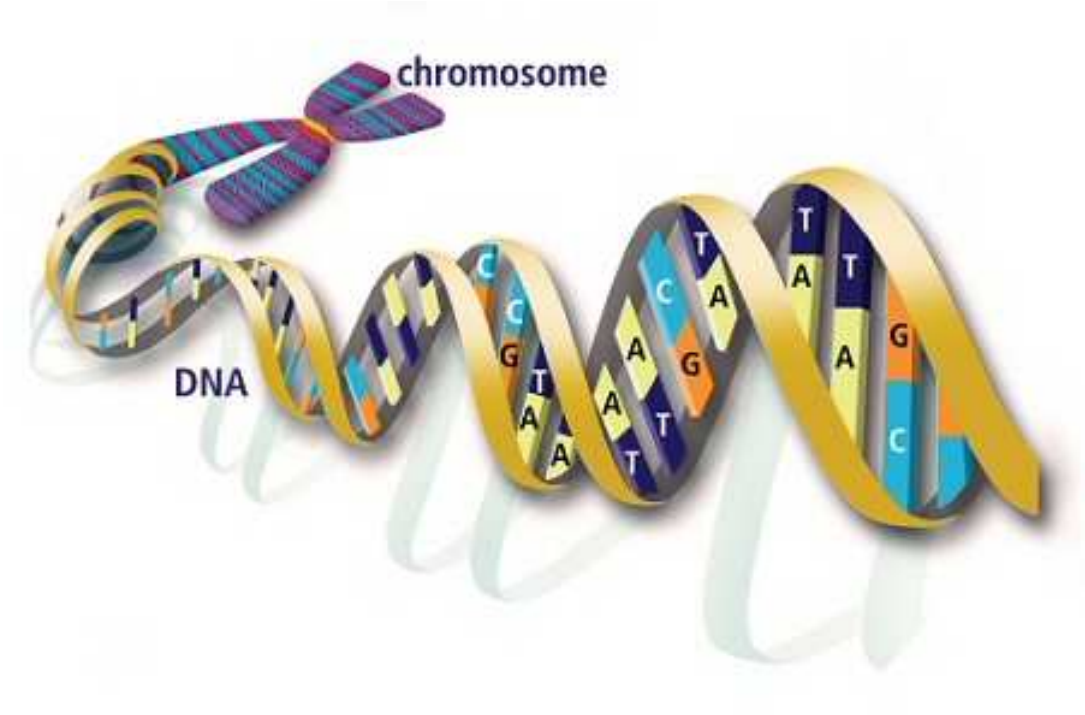
2.1 DNA

Nositelkou genetické informace u většiny organismů je kyselina deoxyribonukleová (DNA). U některých nebuněčných organismů má tuto roli kyselina ribonukleová (RNA) [1].

DNA má tvar dvoušroubovice (obrázek 2.1), která je tvořená dvěma řetězci nukleotidů. Jednotlivé nukleotidy se skládají ze tří složek: organické báze, pětiuhlíkatého cukru a kyseliny hydrogenfosforečné (fosfátu) [1,2].

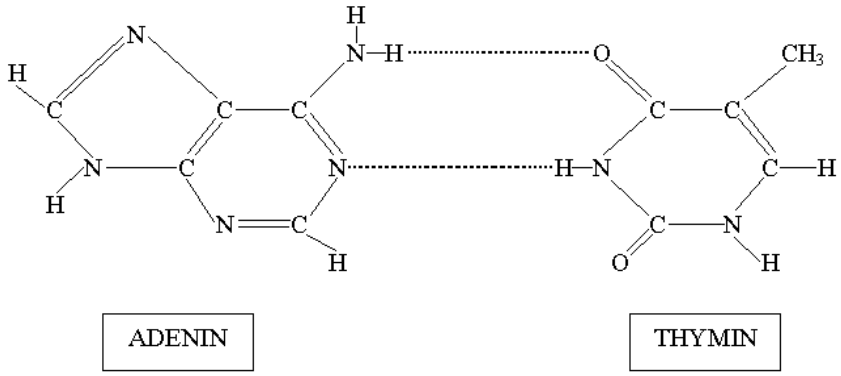
V makromolekule DNA se nacházejí čtyři druhy bází:

- dvě purinové: adenin (A) a guanin (G)
- dvě pyrimidinové: thymin (T) a cytosin (C)

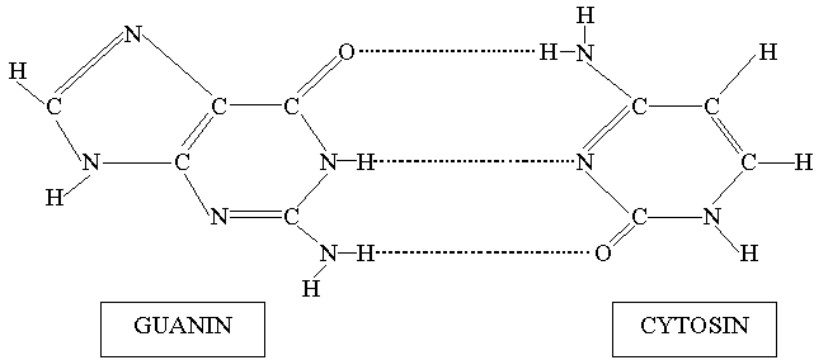


Obrázek 2.1: Dvoušroubovice DNA, která je tvořena nukleotidy. Převzato z [3].

Nukleotidy jsou spojovány pomocí fosfátu a vytváří tak dlouhé nukleotidové řetězce. Molekula DNA je pravotočivá dvoušroubovice, která je tvořena dvěma těmito polynukleotidovými řetězci. Tyto dva řetězce jsou k sobě vázány vodíkovými vazbami mezi bázemi. Vzniká komplementární (doplňková) báze adenin-thymin, které jsou spojeny dvěma vodíkovými můstky (obrázek 2.2) a cytosin-guanin, které jsou spojeny třemi vodíkovými můstky (obrázek 2.3). To znamená, že vlákna jsou komplementární a uchovávají stejnou informaci. Důležité také je pořadí nukleotidů v řetězci, které je základem pro přenos genetické informace [2,4].



Obrázek 2.2: Adenin a thymin jsou vzájemně spojeni pomocí dvou vodíkových můstků. Převzato z [4].



Obrázek 2.3: Guanin a cytosin jsou vzájemně spojeni pomocí tří vodíkových můstků. Převzato z [4].

2.2 Sekundární struktura

Primární struktura je charakteristická pro každý organismus a je dána pořadím a typem nukleotidů. Primární struktura se dá znázornit jako řada nukleotidů a zapsat do řady písmen, které odpovídají jednotlivým nukleotidům (A, C, G a T).

Makromolekuly DNA zaujímají charakteristické prostorové uspořádání, které popisuje jejich sekundární struktura. Sekundární struktura zachycuje tvar a počet řetězců nukleových kyselin.

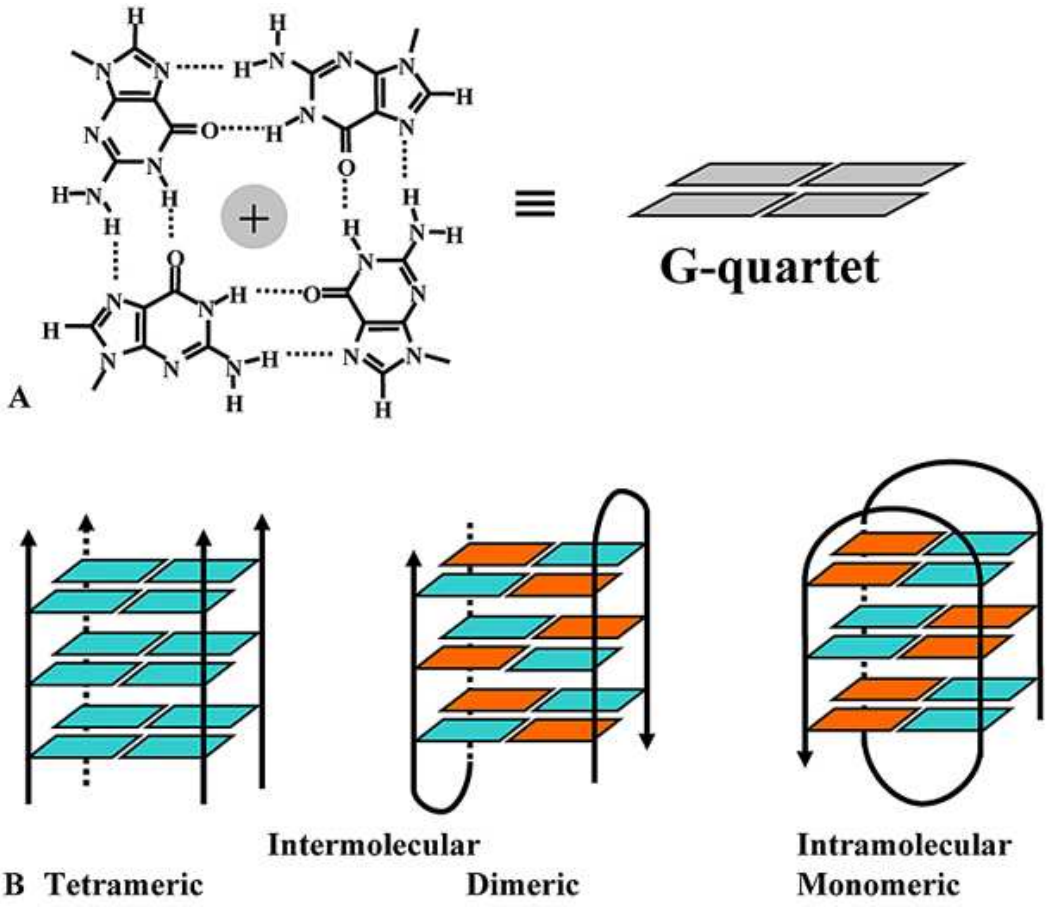
Kromě klasické dvouřetězcové formy stočené do šroubovice, existují i jiné formy molekuly DNA. Například levotočivá dvoušroubovice, ohyb, křížová struktura atd. Pokud se zaměříme na počty řetězců, existují struktury jako triplexy, kvadruplexy či pentaplexy. Tato bakalářská práce se zabývá kvadruplexy, které jsou podrobně popsány níže.

2.3 Kvadruplexy

Mezi nejzajímavější molekuly DNA patří kvadruplexy, které se také označují jako tetraplexy, G4 DNA nebo guaninové kvadruplexy. Tato struktura byla objevena v 60. letech 20. století, ale díky moderní technologii se zvýšil zájem o ně až v uplynulých 10 letech. Kvadruplexy jsou zajímavé pro jejich biologický význam, kdy hrají roli při replikaci a transkripci, a také jsou součástí lidských telomerů. Kromě medicíny jsou kvadruplexy zajímavé i pro nanotechnologii [6].

Nukleové sekvence, které jsou bohaté na guanin, jsou schopny vytvořit kvadruplex. Jeho základní stavební jednotkou je planární uskupení čtyř guaninů (tzv. guaninová tetráda, která je tvořena ze čtyř guaninů vázaných cyklicky) a propojených pomocí vodíkových můstků (vazeb). Ve středu každé tetrády je volný prostor, který je záporně nabitý. Z prostorového pohledu jsou tetrády navrstveny nad sebou a volný prostor tvoří kanál zaplněný ionty [6,7].

Kvadruplexy jsou velmi rozmanité struktury, a proto je lze dělit pomocí mnoha vlastností. Základní rozdělení je podle počtu molekul DNA, podle orientace jednotlivých řetězců a také podle počtu tetrád, který se většinou pohybuje mezi dvěma a čtyřmi [6].



Obrázek 2.4: Uspořádání guaninových bází v tetrádě (A), tetramolekulární, bimolekulární a unimolekulární kvadruplexová struktura (B). Převzato z [5].

V závislosti na počtu molekul DNA, které kvadruplex tvoří, existují kvadruplexy (obrázek 2.4):

- **Unimolekulární** jsou tvořeny jednou molekulou (jedno vlákno DNA).
- **Bimolekulární** jsou tvořeny dvěma molekulami DNA a každé toto vlákno přispívá dvěma guaniny do tetrády.
- **Tetramolekulární** jsou tvořeny čtyřmi nezávislými vlákny DNA a každé toto vlákno přispívá jedním guaninem do tetrády. Tyto čtyři vlákna jsou vždy paralelní.

Unimolekulární a bimolekulární kvadruplexy musí obsahovat smyčky, což jsou spojovací segmenty. Tyto smyčky se mohou skládat s libovolných nukleotidů včetně guaninu. Smyčky mohou být diagonální, laterální a externí. U smyček nás také zajímá jejich délka, sekvence a geometrie [6,7].

Co se týče vzájemné orientace jednotlivých řetězců, můžeme kvadruplexy rozdělit na **paralelní** (typické pro tetramolekulární kvadruplexy) a **antiparalelní** (pokud alespoň jedna smyčka je obrácená vzhledem k ostatním).

Kvadruplexy jsou tedy velmi rozmanité a mohou nabývat velký počet různých topologií, ale základ zůstává stejný, a to několik guaninových tetrád navrstvených nad sebe a volný prostor mezi nimi zaplněný ionty. Tato část je velmi stabilní. Další důležitou vlastností kromě stability je flexibilita, kterou způsobují flexibilní smyčky. Díky tomu mohou kvadruplexy existovat v mnoha topologiích [6,7].

2.3.1 Telomerické kvadruplexy

Tandemové sekvence s vysokým obsahem guaninu, které jsou schopné vytvořit kvadruplexy, se nacházejí v telomerách, které se vyskytují na koncích chromozomů. Telomery jsou opakující se sekvence, které ochraňují chromozom před nežádoucím ovlivňováním s jinými chromozomy. V lidských buňkách jsou tvořeny sekvencí TTAGGG. Tyto dvouřetězcové útvary mají na konci jednovláknový přesah bohatý na guanin. Při stárnutí dochází k dělení buněk, a tím dochází ke zkracování telomer DNA. To ale neplatí pro nádorové buňky, kde ke zkracování nedochází. Ve většině nádorových buněk se totiž nachází enzym telomeráza, který je odpovědný za udržování délky telomer. Telomery a telomeráza je v současnosti velmi důležitý cíl pro výzkum léčiv proti rakovině [6,7,8].

2.3.2 Netelomerické kvadruplexy

V poslední době vzrostl zájem o kvadruplexy, které se nacházejí na jiném místě než v telomerách. Bioinformatické analýzy zjistily, že v genomu existují tisíce sekvencí bohatých na guanin, které by potencionálně mohly tvořit kvadruplexy [8].

2.3.3 Využití kvadruplexů v praxi

Kromě využití v medicíně, kde mají kvadruplexy velký potenciál pro vývoj léčiv proti rakovině, mohou být velmi zajímavé i pro techniku. Mohou být použity jako předlohy pro syntézu nanostruktur a biomateriálů. Mohou sloužit také pro cílené vychytávání nebezpečných kationů. Budoucnost mají také v oblasti biosenzorů, kde mohou být využity jako optické a elektronické senzory. Kvadruplexy mohou být použity i jako nanomotory. Principem je, že se cyklicky utváří a rozplétá kvadruplex. Zajímavé jsou také G-dráty, což jsou dlouhá vlákna (řetězce) složená z kvadruplexů a tyto vlákna mohou obsahovat až tisíce guaninů. Lze toho využít u elektronických nanozařízení, protože obsahují kanál, kde se mohou pohybovat kationy, které jsou nositelem náboje [6].

2.4 Databáze DNA sekvencí

V současné době existuje mnoho obecných a specializovaných biologických databází, která obsahují obrovské množství dat, jež slouží jako důležitý nástroj vědců pro uchovávání znalostí. Tato podkapitola se zabývá databázemi primárních sekvencí DNA, které uchovávají sekvence nukleotidů. Jsou zde uloženy například lidské geny, rostlinné geny, geny různých živočichů a mnoho dalších genů. Tři největší databáze nukleových kyselin jsou EMBL, GenBank a DDBJ [9].

- **EMBL** (European Molecular Biology Laboratory) je hlavní evropská databáze nukleotidových sekvencí a je spravovaná EBI (European Bioinformatics Institute).
- **GenBank** je spravovaná NCBI (National Center for Biotechnology Information) a sídlo je v USA. Tato databáze je nejrozsáhlejší a roste exponenciálně.
- **DDBJ** (DNA Data Bank of Japan) je spravovaná National Institute of Genetics. Tento institut má sídlo v Japonsku.
- **INSDC** (International Nucleotide Sequence Database Collaboration) zajišťuje denní výměnu dat mezi výše uvedenými databázemi.

2.5 FASTA formát

V bioinformatice se FASTA formát (obrázek 2.5) používá k reprezentaci sekvencí nukleotidů. Každý nukleotid v tomto textovém formátu reprezentuje jedno písmeno (A,C,G,T). Tento formát je velmi jednoduchý a velmi často se používá hlavně kvůli jednoduchosti zpracování.

Sekvence ve formátu FASTA je reprezentována řádky, kde délka by neměla překročit 120 znaků, ale obvykle nepřekračuje 80 znaků. První řádek každé sekvence začíná znakem ‘>’ a značí hlavičku dané sekvence, která obsahuje informace o této sekvenci. Pro komentáře je využívá znak ‘;’. Komentáře se ale nepoužívají, jelikož všechny informace nese hlavička. Při zpracovávání se bílé

znaky (mezera a konec řádku) ignorují. V jednom souboru může být více sekvencí, které se oddělují pomocí hlavičky [10].

```
>HSBGPG Human gene for bone gla
TCTTTTTTTCAAGATAGTTTATCTTTAAGTGTTTTTTTTATTAAATAGAAATTTTTGTTTTTATTA
AGGGTAATAAAACTTTATATATATATATATATATATATATATATATATATATATATATATATATAT
ATATATATATATAGAGAAAGAAATTCTCTAATAATTAGAGCCTTGATGGTGAAATGGTAGACACGC
GAGATTCAAAATTTCGTGCTTAAAGCATGGAGGTTCGAGTCCTCTTCAAGGCAATAAAATAAAATT
ATTTAGTTAAATTTTTTAGATAAATATTTTTTATGTTATACTATTCTATTGATAGTAAAGAAATTT
ATATATGTAAAACATATATTTTTTATGA
>HSGLTH2 Human theta 1-globin gene
AAAGGATTTGCAGTCCTCTGCCTTACCACTTGGCCATGTCGCCTTTATTTTAATTAATTGTATAAT
ACAATGTATTATTTTAAAATTCAAGGCATATTCAATATTTTTTATAACAAAATATATTCAATTTTT
ATTTCTAATAATTAATAAAATACACTCCAATAATAATTTAGAGGAAAATATGGAGATTTTTATACT
CCCTGAATTTGGTAAAATACAATTTGAAGGATTTAATCGTTTTATAAATCAAGGTTTGAGTGAAGA
ACTTAGTAATTTTCCAATAATTGAAGATATAGATCAAGAATTCGAGTTTCAAATATTTGGTGAACA
ATATAAATTAGCAGAACCATTATTAAAAGAAAGAGATGCC
```

Obrázek 2.5: Ukázka FASTA formátu.

3 Problematika vyhledávání

3.1 Vyhledávací pravidlo

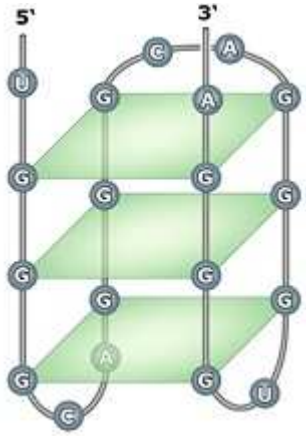
Vyhledávání sekvencí, které mají schopnost vytvořit kvadruplex, je velmi důležité pro další pochopení jejich biologické role v organismech. Kvadruplexy se mohou nacházet v celém genomu jak v telomerách, tak v netelomerických sekvencích.

Pravidlo pro vyhledávání potencionálních unimolekulárních kvadruplexů se řídí podle kritérií:

$$G_{3\text{-}5}N_{1\text{-}7}G_{3\text{-}5}N_{1\text{-}7}G_{3\text{-}5}N_{1\text{-}7}G_{3\text{-}5}\,, \tag{3.1}$$

kde N jsou smyčky, které se mohou skládat z libovolných nukleotidů včetně guaninu, a G je sekvence guaninů (G-skupina). Délka smyček se zpravidla omezuje od jedné do sedmi, protože pokud je smyčka delší, tak je pravděpodobnost vzniku kvadruplexu velmi nízká. Délka sekvence guaninů se většinou omezuje na tři až pět. Pomocí těchto jednotlivých délek lze specifikovat vyhledávací pravidlo [7].

Na obrázku 3.1 je názorně vidět, že unimolekulární kvadruplexy musí mít tři smyčky (mezery), které spojují čtyři G-skupiny, mezi nimiž se nachází právě tyto tři smyčky. Jakmile se tato struktura rozplete, vzniklá sekvence odpovídá vyhledávacímu pravidlu.



Obrázek 3.1: Unimolekulární kvadruplexová struktura. Převzato z [11].

3.2 Problémy při vyhledávání

Na první pohled se může zdát, že vyhledávání kvadruplexů podle daného pravidla je jednoduché, ale existuje zde celá řada problémů, které toto vyhledávání značně komplikuje. V této kapitole jsou tyto problémy vysvětleny a ukázány na konkrétních příkladech.

Mezery obsahující guaniny

Tento problém znázorňuje obrázek 3.2. Z obrázku je zřejmé, že mezeru může tvořit i guanin nebo v horším případě i více guaninů, které se jeví jako G-skupina, která ale nevytváří žádné vazby.

(A) GGGATAGGGAGAGGGTTGGG

(B) GGGAGGGAGGGTTGGGCCGGG

Obrázek 3.2: V mezeře může být guanin (A), v mezeře může být i G-skupina (B).

Překrývání kvadruplexů navzájem

Při použití pravidla pro vyhledávání může vzniknout delší sekvence, která obsahuje více G-skupin, a tím vzniká problém překrývání (obrázek 3.3). Pokud je sekvence dostatečně dlouhá, lze ji rozdělit na dvě či více nepřekrývající se sekvence. Studie zjistily, že počet nepřekrývajících potenciálních kvadruplexů v lidském genomu je asi 375 000 a počet překrývajících kvadruplexů se pohybuje kolem 5 713 000 [7]. Rozdíl mezi těmito dvěma hodnotami je poměrně velký a lze z toho usoudit, že k překrývání dochází poměrně často.

(A) GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG
GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG
GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG
GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG
GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG

(B) GGGAAGGGATAGGGTTGGGCCGGGATAGGGATAGGGTTGGG

Obrázek 3.3: Překrývající kvadruplexové sekvence (A), v některých případech lze sekvenci rozdělit na nepřekrývající sekvence (B).

G-skupina má více možností

Pokud například hledáme kvadruplexy s velikostí G-skupiny tři, ale v sekvenci se vyskytují delší G-skupiny, vzniká problém, kdy je možné nalézt víc možností (obrázek 3.4 A). Z obrázku 3.4 B vyplývá, že pokud jsou délky všech G-skupin delší než minimální délka G-skupiny, výsledný kvadruplex může mít různou délku G-skupin.

(A) GGGATAGGGGACAGGGTTGGG
GGGATAGGGGACAGGGTTGGG

(B) GGGGATAGGGGACAGGGGTTGGGG
GGGGATAGGGGACAGGGGTTGGGG

Obrázek 3.4: Kvadruplexové sekvence se mohou vyskytovat ve více možnostech (A), G-skupiny mohou mít různou velikost (B).

G-skupina je velmi dlouhá

Pokud je sekvence po sobě jdoucích guaninů delší než dvojnásobek minimální velikosti G-skupiny, vzniká komplikace s výskytem všech výše zmíněných problémů (obrázek 3.5 A,B). Jestliže je však sekvence guaninů jdoucích po sobě ještě větší, vzniká nejhorší varianta pro vyhledávání kvadruplexů (obrázek 3.5 C). Počet různých možností výskytu, které mohou nastat, je maximální. V takovýchto případech většinou nemá smysl analyzovat takovou guaninovou sekvenci.

(A) GGGGGGGGAGGGTTGGGCCGGG
GGGGGGGGAGGGTTGGGCCGGG
GGGGGGGGAGGGTTGGGCCGGG
GGGGGGGGAGGGTTGGGCCGGG

(B) GGGGGGGGGGGAAAGGGAAGGG
GGGGGGGGGGGAAAGGGAAGGG

(C) GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG

Obrázek 3.5: Pro delší G-skupiny existuje mnoho možností, jak vytvořit kvadruplexovou strukturu (A, B), ukázka dlouhé guaninové sekvence (C).

Komplementární vlákno

Jak už bylo řečeno, molekula DNA je složena ze dvou vláken, které tvoří dvoušroubovici. Tyto vlákna jsou navzájem komplementární (obrázek 3.6) a při hledání je třeba na to brát ohled. To znamená, že se musí hledat i C-skupiny, protože guanin a cytosin jsou komplementární báze a potenciální kvadruplex se může objevit i v druhém vlákně.

CCCTATCCCTTTCCCACCC

GGGATAGGGAAAGGGTGGG

Obrázek 3.6: Ukázka komplementárních vláken.

4 Analýza a návrh

Cílem této bakalářské práce bylo navrhnout a implementovat vhodný algoritmus pro vyhledávání kvadruplexů. Tato kapitola je zaměřena na návrh, požadavky a popis aplikace.

4.1 Požadavky na aplikaci

Jelikož se při vyhledávání sekundárních struktur v DNA sekvencích pracuje s velkými daty, běžně v řádech megabajtů, ale mohou být i větší (až gigabajtové sekvence), aplikace musí být efektivní. To je třeba zajistit vhodným návrhem algoritmu pro vyhledávání a dobrou implementací. Hlavním problémem je to, že pokud je vyžadováno vyhledání všech kombinací potenciálních kvadruplexů, které mohou nastat, tak vzhledem k různým překrýváním a různým délkám G-skupin nelze tento problém jednoduše a efektivně řešit. Pro efektivní hledání se musí vyhledávat delší úseky, které splňují vyhledávací pravidlo. Následně tyto jednotlivé úseky lze podrobně analyzovat.

Dalším požadavkem je účelnost aplikace. To znamená, že musí provádět správně svoji činnost a také mít srozumitelné a použitelné výsledky. Nabízí se více možností, jak by výstupy mohly vypadat. Nakonec nelze zapomínat na volitelné parametry, kterými lze vyhledávání specifikovat.

4.2 Nástroje pro vyhledávání kvadruplexů

V této podkapitole jsou popsány existující nástroje, jejich možnosti, výstup, výhody a nevýhody. V současnosti existují tři nejznámější vyhledávací programy (Quadparser, QuadFinder a QGRS Mapper), které jsou dostupné přes webové rozhraní.

Quadparser

Quadparser umožňuje nastavit parametry pro vyhledávání jako minimální a maximální velikost smyčky, minimální délku sekvence guaninů a minimální počet guaninových sekvencí.

Quadparser je jednoduchý a efektivní nástroj pro základní vyhledávání sekvencí. Vyhledává sekvence, které odpovídají zadaným parametrům. Výstup programu je podobný jako u mé aplikace pro rychlou analýzu. Výhodou tohoto nástroje je jednoduchost, efektivita a také umí vyhledávat i v komplementárním vláknu.

QuadFinder

Tento nástroj umožňuje nastavit parametry jako minimální a maximální velikost smyčky a minimální a maximální délku sekvence guaninů.

QuadFinder má lehce pokročilejší vyhledávání a zobrazuje překrývající sekvence, ale všechny problémy neřeší a také není schopný vyhledat všechny možné kombinace výskytů, což je jeho nevýhoda. Předností je určitě pěkný grafický výstup a také umí spočítat počet jednotlivých nukleotidů ve vyhledávané sekvenci.

QGRS Mapper

Tento nástroj umožňuje nastavit parametry jako minimální a maximální velikost smyčky, minimální délku sekvence guaninů, maximální velikost potenciální kvadruplexové sekvence a také je možnost zadat nukleotidy, které jsou obsaženy ve smyčce. Z výše zmíněných nástrojů má nejvíce možností nastavení.

QGRS Mapper nabízí mnoho funkcí a jako jediný z uvedených nástrojů je schopen vyhledat všechny kombinace potenciálních kvadruplexů. Zajímavá funkce je ohodnocení každé kombinace podle pravděpodobnosti vzniku a stability. Umí také vybrat ze všech možností nepřekrývající kvadruplexy, kde bere v úvahu jejich hodnocení. Umožňuje také grafické zobrazení výsledné sekvence. Nevýhodou jsou vyšší výpočetní nároky, protože při hledání používá hrubou sílu.

4.3 Popis aplikace

Návrh aplikace vychází z již už existujících nástrojů, jejich vlastností a výstupů, které jsou důležité a neměly by v mé aplikaci chybět. Aby byla zajištěna dostatečná efektivita a uživateli umožněna hlubší analýza pro něho zajímavého úseku sekvence, vyhledávání je rozděleno na dva programy. První program provádí rychlou a efektivní analýzu vstupní sekvence a na výstup vypíše jednotlivé sekvence, které odpovídají pravidlu specifikovaném zadanými parametry. Druhý program poté zpracuje už jen jednu námi zadanou podsekvenci a provede hloubkovou analýzu a zjistí všechny možné potenciální kombinace, které mohou tvořit kvadruplex a ohodnotí je.

4.3.1 Program pro rychlou analýzu

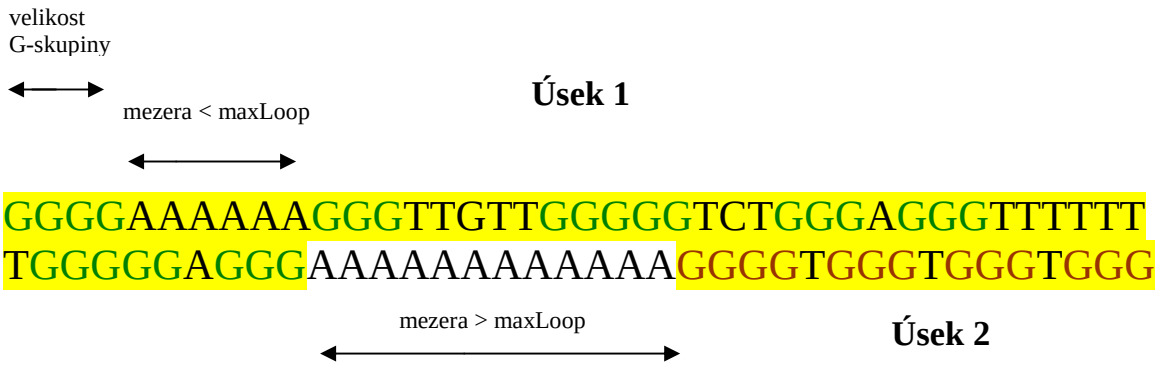
Jak je zmíněno výše, tento program provádí rychlou a efektivní analýzu. Vstupní sekvence je očekávána ve FASTA formátu a může obsahovat i více sekvencí, které jsou oddělené a charakterizované svou hlavičkou. Vyhledávání se provádí podle již výše zmíněného vzorce $G_X N_L G_X N_L G_X N_L G_X$, kde X je délka G-skupiny a L je délka mezery (smyčky). Tyto parametry je možné uživatelsky nastavovat. Nastavují se parametry: minimální a maximální délka, minimální a maximální velikost G-skupiny, maximální velikost kvadruplexu a také báze, ze které je složena G-skupina (většinou G, C). Jestliže nejsou parametry nastaveny, použijí se implicitní hodnoty (tabulka 4.1), které se běžně používají.

`minLoop = 1`	minimální délka mezery
`maxLoop = 7`	maximální délka mezery
`minBasesRepeats = 3`	minimální velikost G-skupiny
`maxBasesRepeats = 5`	maximální velikost G-skupiny
`maxWindow = 35`	maximální velikost kvadruplexu
`basesConsidered = "GC"`	báze, ze kterých se skládá G-skupina

Tabulka 4.1: Implicitní hodnoty parametrů.

Aplikace vyhledává podle vzorce a zadaných parametrů tak, že vyhledá úseky (obrázek 4.1), které splňují tyto podmínky:

- Jednotlivé G-skupiny mají danou délku určenou parametry.
- Mezi těmito G-skupinami musí být mezera odpovídající zadaným parametrům.
- Minimální počet G-skupin musí být takový, aby šel vytvořit potenciální kvadruplex, tedy alespoň čtyři. Musíme brát ale v úvahu, že delší G-skupina může mít v sobě více G-skupin.
- Maximální počet G-skupin je tak velký, dokud není mezera mezi těmito G-skupinami větší než maximální mezera zadaná v parametru.

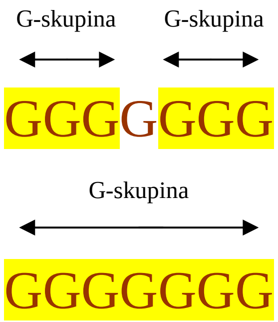


Obrázek 4.1: Reprezentuje vstupní sekvenci, která je rozdělena na úseky v závislosti na velikosti mezery.

Z těchto podmínek je patrné, že se neprovádí vyhledávání, ale že se vstupní sekvence rozdělí na úseky, ve kterých se mohou vyskytovat kvadruplexy. Vyhledané úseky mohou mít i delší velikost, a proto k těmto úsekům jsou uvedeny i doplňující informace:

- Počet G-skupin bez započtení toho, že G-skupina může mít v sobě i více G-skupin.
- Počet G-skupin se započtením vnitřních G-skupin (obrázek 4.2).

- Počet nepřekrývajících se kvadruplexů. Je možno si je představit jako počet dílů, na které se dá výsledný úsek rozdělit, aby se získaly nezávislé (nepřekrývající se) kvadruplexy.



Obrázek 4.2: G-skupina se započtením a bez započtení vnitřních G-skupin.

Ke každému úseku se samozřejmě vypíše pozice výskytu ve vstupní sekvenci, jeho délka a báze, která tvoří G-skupinu. Tato báze se zadává jako vstupní parametr. Normálně se G-skupina skládá z guaninu (G), ale pokud chceme prohledat i druhé vlákno DNA sekvence, tak zadáme i cytosin (C).

```
Parametry: 1 7 3 5 35 CG

================================================================
BLOK 0
hlavicka: Sekvence 1
================================================================
Sekvence cislo: 0
pozice: 298 delka: 79 baze: C
quadruplex: CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCC
pocet G-skupin: 1 relativnich G-skupin: 25 neprekryvajicich: 6
0 79 |

Sekvence cislo: 1
pozice: 390 delka: 38 baze: C
quadruplex: CCCCAAAGGGCCCTTTTTCCCCCCTTTCCCCCCCCCC
pocet G-skupin: 4 relativnich G-skupin: 6 neprekryvajicich: 1
0 4 | 10 3 | 18 6 | 27 11 |

Sekvence cislo: 2
pozice: 446 delka: 30 baze: C
quadruplex: CCCCAAAGGGCCCTTTTTCCCCCCTTTCCC
pocet G-skupin: 4 relativnich G-skupin: 4 neprekryvajicich: 1
0 4 | 10 3 | 18 6 | 27 3 |
```

Obrázek 4.3: Ukázka výstupu programu.

4.3.2 Program pro hloubkovou analýzu

Tento program umožňuje hloubkovou analýzu konkrétní jedné sekvence získané z předchozího programu pro rychlou analýzu. Jako vstup programu se použije výstup z předešlého programu (obrázek 4.3). Jako vstupní parametr se uvede číslo sekvence, která se bude analyzovat, a program si poté sám načte všechny potřebné údaje pro tuto konkrétní sekvenci z výstupu předchozího programu. Tento program používá stejné parametry jako předchozí program a také si načte některé informace o dané sekvenci (pozici, délku, danou sekvenci, počet G-skupin, počet relativních G-skupin a jednotlivé pozice G-skupin), aby je nemusel znovu počítat.

Výstup programu (obrázek 4.4) obsahuje všechny možné výskyty potenciálních kvadruplexů z dané sekvence. Kromě výsledného potenciálního kvadruplexu se zobrazuje i daný kvadruplex ve zkráceném zápisu, jeho pozice, délka a jeho ohodnocení. Toto ohodnocení znázorňuje pravděpodobnost vzniku a stability. Čím větší je toto ohodnocení, tím větší je jeho stabilita a pravděpodobnost vzniku.

```
pozice:0 delka:16 ohodnoceni:64
GGG GG GGG G GGG G GGG
G3 N2 G3 N1 G3 N1 G3

pozice:0 delka:17 ohodnoceni:63
GGG GGG GGG G GGG G GGG
G3 N3 G3 N1 G3 N1 G3

pozice:0 delka:18 ohodnoceni:62
GGG GGGG GGG G GGG G GGG
G3 N4 G3 N1 G3 N1 G3

pozice:0 delka:19 ohodnoceni:62
GGG GGGGG GGG G GGG G GGG
G3 N5 G3 N1 G3 N1 G3
```

Obrázek 4.4: Ukázka výstupu programu.

5 Algoritmus

V této kapitole je popsán princip použitých algoritmů pro každý program, jejich časová složitost a použité datové struktury. Tyto algoritmy nevycházejí z existujících aplikací, protože jejich algoritmy nejsou nikde zveřejněny.

5.1 Rychlá analýza

5.1.1 Algoritmus

Pseudokód

Všechny pseudokódy obsažené v této práci jsou zjednodušené a obsahují hlavní princip algoritmu bez implementačních detailů.

```
//vyhledání všech G-skupin a jejich uložení do pole
for (i=0 ; i < N ; i++) //projde vstupní sekvenci
  if (znak[i] == 'G')
    nBase++
  // počet za sebou jdoucích guaninů >= min. délka G-skupiny
  else if (nBase >= minBasesRepeats)
    // Nalezena G-skupina; uložení pozice a délky do pole G-skupin.
    group[GIndex].position = i – nBase
    group[GIndex].countG = nBase
    nBase = 0
    GIndex++
  else
    nBase = 0

//průchod polem G-skupin a vyhledání kvadruplexů
for (i=0 ; i < GIndex – 1 ; i++)
  gap = mezera_mezi_G-skupinami()
  if (gap <= maxLoop && gap >= minLoop) //pokud mezera vyhovuje
    nGGroup++
    //počet relativních G-skupin
    relativeBases += NumberOfRelative(group[i+1].countG)
  else if (nGGroup >= 3 || relativeBases >= 4)
    // je nalezen qudruplex
    tisk_vseho_potrebneho_na_vystup()
    nGGroup = 0
    relativeBases = NumberOfRelative(group[i+1].countG)
  else
    nGGroup = 0
    relativeBases = NumberOfRelative(group[i+1].countG)
```

Slovní popis

Algoritmus pro rychlou analýzu DNA sekvence je navržen co nejefektivněji, protože vstupní sekvence mohou být velmi dlouhé. Důraz je kladen na to, aby výstupy byly pro uživatele dobře použitelné a srozumitelné.

Princip tohoto algoritmu je takový, že se projde lineárně vstupní sekvence a hledají se G-skupiny (skupiny po sobě jdoucích guaninů), které mají délku větší než minimum zadané parametrem programu. Pokud se taková skupina najde, uloží se začátek této G-skupiny a její délka do pole. Následně se lineárně prochází toto pole s G-skupinami a zjišťuje se, zda je mezera mezi nimi menší než maximální dovolená mezera a větší než minimální dovolená mezera. Tyto mezery jsou určeny parametry. Aby byla nějaká sekvence označena za sekvenci, která může vytvořit kvadruplex, musí být za sebou alespoň čtyři G-skupiny. Je třeba brát v úvahu, že některé delší, po sobě jdoucí guaninové sekvence obsahují více G-skupin, a ty je potřeba započítat. Délka vyhledávané sekvence může být tak dlouhá, dokud není mezera mezi dvěma G-skupinami větší než maximální mezera.

Kromě samotného vyhledávacího algoritmu bylo třeba vyřešit i další věci, například rozdělení vstupního souboru (pokud obsahuje více sekvencí) podle hlavičky do pole, které obsahuje jednotlivé sekvence. Tyto bloky se pak postupně projdou pomocí výše zmíněného vyhledávacího algoritmu. Dále bylo třeba odstranit ze vstupní sekvence bílé znaky (hlavně znak konce řádku) a všechny znaky převést na velká písmena. Pokud je zadáno prohledávání i druhého vlákna vstupní DNA sekvence, je třeba jako parametr zadat C (cytosin). V programu to bude znamenat, že se postupně budou hledat G-skupiny obsahující G (guanin) a poté se provede další průchod, který bude hledat G-skupiny s C (cytosin). Lze vyhledávat i G-skupiny obsahující thymin nebo adenin, takže jako parametr se zadá T a A, ale v praxi je to zbytečné.

5.1.2 Časová složitost

V tomto případě lze časovou složitost programu vyjádřit poměrně dobře. Zde lze vyhledávání rozložit na dva podproblémy. První z nich je lineární průchod vstupní sekvencí, kdy se hledají G-skupiny, které se následně ukládají do pole. Z toho vyplývá, že časová složitost je lineární. U druhého podproblému se pak lineárně projde pole s G-skupinami a vyhledají se potenciální kvadruplexy. Opět se jedná o lineární časovou složitost.

5.1.3 Datové struktury

Při návrhu a implementaci je dobré použít vhodné datové struktury, které budou srozumitelně reprezentovat dané problémy a zpřehlední výsledný zdrojový program.

V prvním programu, který je určen pro rychlou analýzu, jsou použity struktury SChunk a SGGroup. Struktura SGGroup reprezentuje G-skupinu a obsahuje pozici dané G-skupiny a její délku. Více informací není třeba uchovávat. V samotné aplikaci je vytvořeno pole těchto struktur a při

lineárním průchodu vstupní sekvence se postupně naplní jednotlivými G-skupinami. Struktura SChunk reprezentuje podsekvence (dále jen relativní kvadruplexy), které splňují podmínku pro vytvoření kvadruplexu. Tato struktura obsahuje ukazatel na začátek relativního kvadruplexu, pozici ve vstupní sekvenci, počet G-skupin a délku relativního kvadruplexu. Při vyhledávání se struktura naplní daty, jež se použijí pro výpočet ostatních informací, které budou na výstupu.

5.2 Podrobná analýza

5.2.1 Algoritmus

Pseudokód

```
if (chunk.NumRelative == 4) // počet relativních kvadruplexů je 4
  // lze redukovat kombinace
  redukce()
//Vygenerování kombinací
num = celkovy_pocet_kombinaci()
for(i=0 ; i < num ; i++) //pro všechny kombinace
  for (i=0 ; i < 3 ; i++) //pro všechny mezery
    if (gap[i] == maxLoop) //tato mezera přeteče
      gap[i] = minLoop
    else
      gap[i]++ //zvýšení délky dané mezery
      ulozeni_kombinace()
      break
  if (i == 3) ) //všechny mezery přetekly
    GGroupLen++ //zvýšení délky G-skupiny

for (i=0 ; i < N ; i++) //index projde vstupní sekvenci
  for (j=0 ; j < num ; j++) //pro všechny kombinace
    //porovnání kombinace se vstupní sekvencí
    for (k=0 ; k < 4 ;) //projde všechny G-skupiny
      if (!isInGroup(group[k],sequence[i]))
        //G-skupina neleží v G-skupině sekvence[i]
        break
      k++
    if (k==4)
      //porovnání dopadlo v pořádku; nalezen kvadruplex
      vypocet_ohodnoceni()
      tisk_vseho_potrebneho_na_vystup()
```

Slovní popis

V tomto případě je dobré vymyslet také efektivní algoritmus, ale vzhledem k tomu, že se zjišťují všechny možné výskyty kvadruplexů a k mnoha problémům, které se zde vyskytují, nelze jednoduše tento úkol vyřešit. Jelikož tento algoritmus analyzuje pouze menší sekvence (desítky až stovky bází), na efektivitu jsou kladeny daleko menší nároky než v předchozím případě.

Na první pohled by se dala sekvence, která je výsledkem předchozího algoritmu, rozdělit postupně na čtyři za sebou jdoucí G-skupiny a z těch zjistit všechny možné výskyty. Tento přístup by však nenalezl všechny možnosti, protože mezera může obsahovat i G-skupinu a také G-skupina může obsahovat více G-skupin. Vzhledem k potřebě nalezení všech možností je tento způsob nepoužitelný pro daný problém.

Pro vyhledání všech možností je třeba použít jiný přístup - takový, kdy se vygenerují všechny kombinace podle vstupních parametrů, jež se postupně porovnávají s analyzovanou sekvencí. Porovnává se pouze to, zda všechny G-skupiny z dané kombinace leží v G-skupinách analyzované sekvence. Jelikož vstupní sekvence může být i delší a obsahovat více G-skupin, je třeba index, který ukazuje na začátek vstupní sekvence, posouvat. Pokud se index nachází uvnitř G-skupiny, posouvá se o jedno místo, pokud to má smysl. Jinak se přeskočí o víc míst tak, aby se index dostal na začátek další G-skupiny a zase se provádí porovnání se všemi kombinacemi, dokud to má smysl (zbývající velikost vstupní sekvence je větší než minimální velikost kvadruplexu).

5.2.2 Generování kombinací

Pro nalezení všech možných kombinací se nejprve vypočítá počet těchto kombinací. U tohoto výpočtu se opět vychází ze vzorce $G_{X1\text{-}X2}\, N_{L1\text{-}L2}\, G_{X1\text{-}X2}\, N_{L1\text{-}L2}\, G_{X1\text{-}X2}\, N_{L1\text{-}L2}\, G_{X1\text{-}X2}$.

Pokud mají všechny mezery stejnou délku, výpočet vypadá:

$$N_{COM} = (L_2 - L_1 + 1)^3 \cdot (X_2 - X_1 + 1) \qquad (5.1)$$

Jakmile se vypočítá počet kombinací, tak se cyklem postupně inkrementuje kombinace tolikrát, kolik je celkový počet kombinací. Inkrementace se provádí tak, že se postupně zvyšuje první mezera, dokud není větší jak maximální mezera. V okamžiku, kdy je větší, dojde k „přetečení“ a zvýší se druhá mezera. Jakmile přeteče druhá mezera, tak se zase zvýší třetí mezera. Jakmile přetečou všechny, tak se zvýší velikost G-skupiny a inkrementace se provádí od začátku. Tímto způsobem se vygenerují úplně všechny kombinace.

5.2.3 Redukce kombinací

Za určitých podmínek lze generované kombinace redukovat. K redukci dochází v případě, že počet G-skupin se započtením vnitřních G-skupin (dále jen relativní G-skupiny) je čtyři. V takovém případě lze počet výsledných kombinací mnohonásobně snížit. V nejlepším případě se vygenerují pouze kombinace, které všechny vyhoví následnému porovnávání se vstupní sekvencí.

Pokud je tedy počet relativních G-skupin roven čtyři, vznikají případy, které znázorňuje tabulka 5.1. Samotná redukce spočívá v tom, že pro každou mezeru v sekvenci lze vypočítat maximální a minimální mezeru a spočítat maximální velikost G-skupiny.

Poté bude výpočet celkového počtu kombinací vypadat:

$$\mathrm{N}_{\mathrm{COM}} = (L_{12} - L_{11} + 1) \cdot (L_{22} - L_{21} + 1) \cdot (L_{32} - L_{31} + 1) \cdot (X_2 - X_1 + 1), \tag{5.2}$$

kde L_{1X} znamená první mezera, L_{2X} druhá mezera atd. a N je velikost G-skupiny.

Počet G-skupin	Uspořádání relativních G-skupin v sekvenci	Příklad vstupní sekvence
1	4	GGGGGGGGGGGGGGGG
2	1:3	GGGTTTGGGGGGGGGGG
	3:1	GGGGGGGGGGGTTTGGG
	2:2	GGGGGGGTTTGGGGGGG
3	1:1:2	GGGTTTGGGTTTGGGGGGG
	1:2:1	GGGTTTGGGGGGGTTTGGG
	2:1:1	GGGGGGGTTTGGGTTTGGG
4	1:1:1:1	GGGTTTGGGTTTGGGTTTGGG

Tabulka 5.1: Případy, které mohou nastat, pokud je počet relativních G-skupin čtyři.

U všech případů, které zobrazuje tabulka 5.1, se postupuje obdobně s drobnými odlišnostmi. Proto se tyto případy rozdělí podle tabulky. Vznikají dva přístupy, jak vypočítat minimální a maximální velikost mezer. První případ je vidět na posledním řádku tabulky. V tomto případě se minimální mezera rovná velikosti mezery mezi G-skupinami. Maximální mezera se vypočítá tak, že se přičte k minimální mezeře ještě rozdíl velikosti G-skupiny s minimální velikostí G-skupiny. Přičítáme tedy rozdíly obou G-skupin mezi nimiž mezera leží. Tedy:

$$L_{12} = L_{11} + (X_{G1} - X_1) + (X_{G2} - X_1), \tag{5.3}$$

kde X_{G1} je velikost první G-skupiny a X_{G2} je velikost druhé G-skupiny. Ostatní proměnné mají stejný význam jako v předešlých případech.

Druhý přístup znázorňuje první řádek tabulky. To znamená, že G-skupina obsahuje více G-skupin. V tomto případě počet relativních G-skupin je čtyři a všechny se nachází v jedné

G-skupině. Maximální velikost mezery se poté vypočítá tak, že se od délky sekvence odečte čtyřikrát minimální délka G-skupiny a dvakrát minimální délka mezery. Tedy:

$$N - 4 \cdot X_1 - 2 \cdot L_1 , \tag{5.4}$$

kde *N* je velikost vstupní sekvence.

Ostatní případy, které jsou v tabulce, už kombinují oba výše zmíněné přístupy, tudíž se pro výpočet jednotlivých mezer použije vždy vhodný postup, který používá jeden z těchto dvou výše zmíněných principů.

Maximální délka G-skupiny se počítá pro první případ tak, že se vybere nejmenší velikost G-skupiny ze všech G-skupin a tato hodnota se poté označí za maximální. Pro druhý případ se maximální délka spočítá tak, že se postupně zvyšuje velikost G-skupin a zároveň je třeba dodržet podmínku, aby celková velikost kvadruplexu byla menší než velikost vstupní sekvence. Nejvyšší hodnota, která splňuje tuto podmínku, je pak označena jako maximální velikost G-skupiny.

5.2.4 Časová složitost

V tomto případě už nelze vyjádřit časovou složitost jednoduše, protože kromě vygenerovaného počtu kombinací, které závisí na vstupních parametrech, časová složitost závisí také na struktuře a délce vstupní kvadruplexové sekvence. V této podkapitole je popsána časová složitost pro sekvenci, kde lze redukovat kombinace, a také pro sekvenci, kde kombinace redukovat nelze.

Kombinace nelze redukovat

Pokud se berou v úvahu implicitní parametry pro délky mezer a délky G-skupin (1,7,3,5), výpočet počtu kombinací se provede dle vzorce 5.1 takto:

$$N_{\mathrm{COM}} = (L2 - L1 + 1)^3 \cdot (X2 - X1 + 1) = (7 - 1 + 1)^3 \cdot (5 - 3 + 1) = 1029$$

Pokud je celková délka vygenerované kombinace větší než maximální velikost kvadruplexu zadaná pomocí parametru, tak se zahazuje. Pro zjednodušení výpočtu nebude s touto situací uvažováno.

Každá takto vygenerovaná kombinace se musí porovnávat se vstupní sekvencí. Je nutno brát v úvahu, že po každém porovnání všech kombinací je zapotřebí posunout index, který ukazuje na začátek vstupní sekvence, o jedno místo. Tím se výrazně zvýší počet porovnávání.

Nejhorší případ nastává, pokud je sekvence tvořena jen z guaninů. Zde je konkrétní případ, kdy je vstupní sekvenci délky 40:
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG.

Výpočet počtu porovnávání kombinací se vstupní sekvencí vypadá:

$$N_{CEL} = N_{COM} \cdot i\,, \tag{5.5}$$

kde *i* je počet posunutí indexu. Pro tuto sekvenci se vypočítá jako délka vstupní sekvence mínus minimální délka kombinace. Tedy:

$$N_{CEL} = N_{COM} \cdot (N - \min Window) \tag{5.6}$$

Pro tento konkrétní případ je výpočet:

$$N_{CEL} = N_{COM} \cdot (N - \min Window) = 1029 \cdot (40 - 15) = 25725$$

Pro normální sekvence, které obsahují i mezery, není nutné posouvat index vždy o jedno místo. To nastává v případech, kdy se index nachází v G-skupině a zbývající délka této G-skupiny je menší než minimální délka G-skupiny nebo se index nachází v mezeře. V těchto případech se index automaticky posunuje na začátek následné G-skupiny.

Příklad normální vstupní sekvence vhodné pro vyhledávání také délky 40:
GGGGGGGGAAAGGGGAAAGGGGGGGAAAGGGGAAAGGGGGGGG.

Pokud se do programu vloží počítadlo, které ukazuje počet porovnávání, bude ukazovat číslo 13 364. Toto číslo je menší než v předchozím případě, přestože vstupní sekvence má stejnou délku. Je to způsobeno tím, že se index posouvá o více míst než o jedno.

Kombinace lze redukovat

Na redukci počtu kombinací není obecný vzorec, takže výpočet bude předveden na konkrétním příkladu. Vstupní sekvence je:
GGGGAAGGGAGGGAAGGGG.
Ze sekvence je potřeba zjistit parametry pro velikost mezer a pro velikost G-skupin. Použijeme vzorec 5.2 a provedeme redukci takto:

$$N_{\text{COM}} = (L_{12} - L_{11} + 1) \cdot (L_{22} - L_{21} + 1) \cdot (L_{32} - L_{31} + 1) \cdot (X_2 - X_1 + 1) =$$
$$= (3 - 2 + 1) \cdot (1 - 1 + 1) \cdot (3 - 2 + 1) \cdot (1 - 1 + 1) = 4$$

Počet posunů indexu se řídí podle první G-skupiny. V tomto případě stačí posunout index jednou. Takže při výpočtu se bude počet kombinací násobit dvěma. Podle vzorce 5.5 bude výpočet počtu porovnávání se vstupní sekvencí vypadat:

$$N_{CEL} = N_{COM} \cdot i = 4 \cdot 2 = 8$$

Celkový počet porovnávání je mnohem nižší hlavně díky redukci kombinací a také díky malému počtu posunutí indexu.

5.2.5 Ohodnocení kvadruplexu

Každý potenciální kvadruplex je ohodnocen podle schopnosti vytvořit stabilní kvadruplex. Čím vyšší ohodnocení, tím vyšší je jeho schopnost pro tvorbu stabilního kvadruplexu. Při výpočtu ohodnocení se vycházelo z nástroje QGRS Mapper [12]. Vzorec pro výpočet vychází z dobře známých kritérií, které mají hlavní vliv na tvorbu stabilního kvadruplexu:

- Mezery mají tendenci mít zhruba stejnou délku.
- Delší G-skupiny jsou více stabilní.

Vzorec pro výpočet ohodnocení je:

$$G_{RAT} = G_{GLEN} \cdot G_{KOEF} - G_{AVG}, \tag{5.7}$$

kde G_{GLEN} je délka G-skupiny mínus jedna. Tento parametr má největší vliv na celkové ohodnocení. G_{KOEF} je koeficient závislý na maximální velikosti kvadruplexu (zadáno vstupním parametrem) a G_{AVG} je průměrný rozdíl délek mezer. Tento průměrný rozdíl se vypočítá:

$$G_{AVG} = \frac{|y1 - y2| + |y1 - y3| + |y2 - y3|}{3}, \tag{5.8}$$

kde *y1*, *y2* a *y3* jsou délky jednotlivých mezer.

5.2.6 Datové struktury

V druhém programu, který slouží pro hloubkovou analýzu konkrétní sekvence, jsou použity datové struktury SGGroup, SChunk, STypeItem a SType. Struktura SGGroup má stejný význam jako v předchozím programu - reprezentuje G-skupinu. Struktura SChunk také reprezentuje relativní kvadruplex, ale už se trochu liší, protože obsahuje víc informací o daném relativním kvadruplexu potřebných pro vyhledávaní. Kromě již čtyř zmíněných položek tato struktura také obsahuje bázi, ze které je složena G-skupina, počet relativních G-skupin a jednotlivé G-skupiny, z nichž je tvořen tento

relativní kvadruplex. Pole struktur typu SGGroup reprezentuje jednotlivé G-skupiny. Struktura SChunk je naplněna daty ze vstupu, které obsahují všechny potřebné informace pro vyhledávání.

Pro reprezentaci všech vygenerovaných kombinací jsou použity struktury STypeItem a SType. STypeItem má jednu položku - pole, které obsahuje délky mezer a délky G-skupin uložených postupně od začátku. STypeItem reprezentuje tedy jednu kombinaci. Struktura SType uchovává všechny vygenerované kombinace a informace o nich. Obsahuje tedy pole struktur STypeItem, počet G-skupin za sebou (normálně je to číslo čtyři), celkový počet kombinací a počet využitých kombinací. Počet využitých kombinací může být nižší než celkový počet kombinací, protože při generování se sleduje, zda je délka každé kombinace menší než zadaná maximální velikost kvadruplexu.

6 Testování

Tato kapitola se zabývá testováním aplikace pro rychlou analýzu. Zaměřuje se především na porovnání náhodně vygenerované DNA sekvence a reálné sekvence. Obě sekvence mají přibližně velikost 56 MB. To znamená, že obsahují asi 56 miliónů bází.

Při tomto testování se měnila minimální délka G-skupiny od dvou do šesti a zjišťoval se celkový počet nalezených kvadruplexů a maximální počet nepřekrývajících se kvadruplexů pro obě vstupní sekvence. Pro všechna měření byla minimální délka mezery jedna a maximální délka mezery byla sedm.

Výsledky testování lze vidět v tabulce 6.1. Z naměřených hodnot vyplývá, že při minimální délce G-skupiny dva a tři jsou výsledné hodnoty stejného řádu. Zajímavější zjištění už nastává v případě minimální délky G-skupiny čtyři a pět. Náhodně vygenerovaná sekvence nenalezne žádnou kvadruplexovou sekvenci, protože pravděpodobnost tohoto nalezení se vzrůstající délkou G-skupiny exponenciálně klesá. Reálná sekvence ovšem nějaké výskyty nalezne. Lze to vysvětlit tím, že kvadruplexy mají biologický význam. Pro minimální délku G-skupin šest, už nebyly nalezeny žádné možné výskyty kvadruplexů.

Minimální délka G-skupiny	Náhodně vygenerovaná sekvence		Homo sapiens GRCh37 57	
	nalezených kvadruplexů	nepřekrývajících se kvadruplexů	nalezených kvadruplexů	nepřekrývajících se kvadruplexů
2	130 538	132 749	93 180	105 477
3	1 328	1 328	1 452	1 475
4	0	0	76	77
5	0	0	12	12
6	0	0	0	0

Tabulka 6.1: Zjištěné hodnoty při testování.

7 Závěr

Kromě samotného návrhu a implementace algoritmů pro vyhledávání kvadruplexů bylo nutné proniknout do oblastí, které se zabývají DNA, sekundárními strukturami a hlouběji prostudovat samotné kvadruplexy. Také bylo třeba osvojit si některé kapitoly z oboru bioinformatiky, například formáty pro uložení DNA sekvencí či databáze DNA sekvencí.

V rámci této práce byla vytvořena efektivní aplikace pro vyhledávání potenciálních kvadruplexů v DNA sekvencích. Aby byla zajištěna dostatečná efektivita a uživatel měl možnost zvolit si tvar výstupu, který ho zajímá, byly vytvořeny dva programy. První, který provádí rychlou analýzu, je efektivní a rozdělí vstupní sekvenci na úseky, které mohou obsahovat potenciální kvadruplexy. Druhý program pro hloubkovou analýzu pak v dané sekvenci vyhledá všechny možnosti výskytu kvadruplexu. Současné aplikace toto buď neumožňují anebo prohledávají celou vstupní sekvenci, což může být neefektivní.

Na závěr jsou nastíněny další možnosti vývoje projektu. Určitě by bylo dobré vytvořit uživatelské rozhraní nebo webové rozhraní, které by umožňovalo pěkný a přehledný grafický výstup, popřípadě toto rozhraní propojit s některou již existující databází. Mé aplikace by se poté upravily a použily jako CGI skripty.

Literatura

[1] DNA In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2005-01-30, 2009-03-09 [cit. 2010-05-04]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/DNA>.

[2] Nukleová báze In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2009-03-15, 2009-03-15 [cit. 2010-05-04]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Nukleov%C3%A9_b%C3%A1ze>.

[3] Childrensmuseum.org [online]. c2009 [cit. 2010-05-04]. DNA - The Code of Life. Dostupné z WWW: <http://www.childrensmuseum.org/teachers/unitsofstudy/biotechnology/lesson3.htm>.

[4] Genetika.wz.cz [online]. c2009 [cit. 2010-05-04]. DNA & RNA. Dostupné z WWW: <http://genetika.wz.cz/dnarna.htm>.

[5] Bioscience.org [online]. 2007-05-01 [cit. 2010-05-04]. Frontiers in Bioscience. Dostupné z WWW: <http://www.bioscience.org/2007/v12/af/2412/figures.htm>.

[6] ŠPAČKOVÁ, Naděžda. Tři jsou málo, pět je moc aneb seznamte se s kvadruplexy. Živa. 2009, 3, s. 98-100.

[7] Burge S., Parkinson N. G., Hazel P., Todd K. A., Neidle S.: Quadruplex DNA: sequence, topology and structure [online]. [cit. 2010-05-04]. Dostupné z WWW: <http://nar.oxfordjournals.org/cgi/content/short/34/19/5402>.

[8] G-quadruplex In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2006-01-01, 2010-04-06 [cit. 2010-05-04]. Dostupné z WWW: <http://en.wikipedia.org/wiki/G-quadruplex>.

[9] Biological database In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2003-12-08, 2010-04-28 [cit. 2010-05-04]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Biological_database>.

[10] FASTA format In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2004-02-14, 2010-03-18 [cit. 2010-05-04]. Dostupné z WWW: <http://en.wikipedia.org/wiki/FASTA_format>.

[11] Bioinformatics.ramapo.edu [online]. 2006-07-01 [cit. 2010-05-04]. QGRS Mapper. Dostupné z WWW: <http://bioinformatics.ramapo.edu/QGRS/background.php>.

[12] D'Antonio L., Bagga P.: Computational Methods for Predicting Intramolecular G-Quadruplexes in Nucleotide Sequences. CSB, IEEE Computational Systems Bioinformatics Conference (CSB'04) 2004, s. 590-591, ISBN 0-7695-2194-0/04

Seznam příloh

Příloha 1. CD/DVD obsahující:

- Zdrojové texty.
- Soubor navod.pdf obsahující návod k použití programů.
- Testovací skript test.sh pro otestování programů.
- Technickou zprávu bakalářské práce ve formátu PDF v souboru xnemec26_bp.pdf.
- Technickou zprávu bakalářské práce ve formátu DOC v souboru xnemec26_bp.doc.